WE CLAIM:

1. A network management and service provisioning environment comprising a framework, the framework including:

   a. an implementation of a single managed entity object class, the single managed entity object class being run-time derivable via type derivation into a hierarchy of managed entity object types minimizing the need to re-code and re-compile framework software application code in support of new managed entity object types;

   b. a registry for run-time registration of at least one plug-in brokering access to network management and service provisioning enabling technologies;

   c. a parser for processing at least one self-contained managed data network entity specification;

   d. a generic lexical analyzer interpreting at least one directive; and

   e. an interpreter for processing messages received from at least one network management and service provisioning software application

   wherein a separation is achieved between managed entities, enabling technologies and software applications, the separation enabling independent development, maintenance and troubleshooting in providing network management and service provisioning solutions.

2. as claimed in claim 1, wherein the single managed object class is an abstract managed object class.

3. as claimed in claim 1, wherein the derivation of the managed entity object type hierarchy includes the specification of at least one attribute.

4. as claimed in claim 1, wherein the at least one self contained managed data network entity specification includes a human readable file.

5. as claimed in claim 4, wherein the human readable file is an attribute file holding attributes corresponding to a single managed entity object type.

6. as claimed in claim 1, wherein the at least one directive includes an attribute specification.

7. as claimed in claim 6, wherein the attribute specification further specifies managed entity object type inheritance.

8. as claimed in claim 1, wherein the network management and service provisioning enabling technologies include support for at least one of a persistence method and a persistence entity.

9. as claimed in claim 1, wherein the at least one directive further specifies a command sequence to be followed in using a specific registered enabling technology.

10. as claimed in claim 9, wherein the framework further comprises at least one registered enabling technology specific lexical analyzer stub for interpreting at least one enabling technology specific directive.

11. A network management and service provisioning apparatus implementing the network management and service provisioning environment claimed in claim 1.

12. A method of providing a network management and service provisioning solution comprising steps of:

    a. registering with a framework at least one plug-in brokering access to at least one network management and service provisioning enabling technology;

    b. parsing at least one managed data network entity specification loaded by the framework;

c. deriving a single managed entity object class into a managed object type hierarchy via type derivation; and

d. processing at least one message received by the framework from at least one network management and service provisioning software application;

wherein framework acts as an enabler by separating managed data network entities, enabling technologies and software applications, as well as a facilitator therebetween in providing the network management and service provisioning solution.

13. A method as claimed in claim 12, wherein processing the at least one message received by the framework, the method comprises a further step of deriving a containment hierarchy of managed entity object type instances corresponding to field installed data network equipment.

14. A method as claimed in claim 12, wherein registering with the framework at least one plug-in, the method further comprises a step of run-time registering the at least one plug-in.

15. A method as claimed in claim 14, wherein run-time registering the at least one plug-in, the method further comprises a prior step of: selecting the at least one plug-in for registration thereof.

16. A method as claimed in claim 12, wherein parsing the at least one managed data network entity specification loaded by the framework, the method further comprises a step of: run-time loading the at least one managed data network entity specification.

17. A method as claimed in claim 16, wherein run-time loading the at least one managed data network entity specification, the method further comprises a prior step of: selecting the at least one managed data network entity specification.

18. A method as claimed in claim 12, wherein parsing, the method further comprises a step of: extracting at least one directive therefrom, the at least one managed data network entity specification being associated with at least one managed entity object type.

19. A method as claimed in claim 12, wherein deriving a single managed entity object class via type derivation, the method further comprises a step of setting at least one attribute.

20. A method as claimed in claim 12, wherein prior to processing the at least one message received by the framework from the at least one software application, the method further comprises a step of: registering the at least one software application with the framework.

21. A method as claimed in claim 12, wherein processing the at least one message received by the framework; the method further comprises a step of: implementing a directive specified in the at least one managed data network entity specification using a lexical analyzer stub associated with the at least one plug-in.

22. A method as claimed in claim 21, wherein implementing the directive, the method further comprises a step of: instantiating managed entity object types.

23. A method as claimed in claim 21, wherein implementing the directive the method further comprises a step of: effecting a change in a network state of a managed data transport network in a realm of management.

24. A method as claimed in claim 12, wherein subsequent to processing the at least one message received by the framework; the method further comprises a step of: sending a message to the software application.